REMARKS

Applicants appreciate the time taken by the Examiner to review Applicants' present application. Applicants respectfully request reconsideration and favorable action in this case.

I. Hattrup Not Available as a Reference

As an initial matter, Applicants respectfully submit that Hattrup is not prior art under 35 U.S.C. 102(e). Applicant invented the subject matter of rejected Claims 1-23 prior to the effective date of Hattrup. The effective date of Hattrup is the filing date, May 23, 2003. The attached Declaration Under 37 C.F.R. 1.131 established that Applicant invented the subject matter of rejected Claims 1-23 at least as early as July 12, 2002. The Declaration states that Steve Justiss and Rob Sims, employees of Crossroads Systems, Inc. are original joint inventors of the invention described in the present Application. The Declaration further states that as early as July 12, 2002, Robert Sims and Steve Justiss conceived the invention of the present Application. A copy of an invention disclosure form evidencing conception at least as early as July 12, 2002 is attached as Exhibit A to the Declaration. The Declaration further states that Mark Berrier of Gray Cary sent Steve Justiss and Rob Sims a letter including a draft application describing the present Application on February 21, 2003 and that the application was filed on August 7, 2003. A copy of the February 21, 2003 letter is also attached hereto as Exhibit B of the Declaration. Applicants therefore respectfully submit that the date of invention of the present application was prior to the effective date of the Hattrup reference.

II. Rejections under 35 U.S.C. § 103

A. Introduction

Claims 1-23 stand rejected as obvious over U.S. Publication No. 2004/0243736 ("Hattrup") in view of U.S. Patent No. 6,892,199 ("Hong ").

In order to establish a prima facie case of obviousness, the Examiner must show: that the prior art references teach or suggest all of the claim limitations and that there is some suggestion or motivation in the references (or within the knowledge of one of ordinary skill in the

art) to modify or combine the references and that there is a reasonable expectation of success of such combination. M.P.E.P. 2142, 2143; In re Vaeck, 947 F. 2d 488, 20 U.S.P.Q.2d 1438 (Fed. Cir. 1991).

B. Prima Facie Case Fails without Hattrup Fails

The Examiner relies on Hattrup to show where various features of the present invention can be found. As Hattrup is not a proper prior art reference, Applicants respectfully request that the Examiner point out where the features for which the Examiner relied on Hattrup can be found in the Hong reference or allow Claims 1-23.

C. Claims 1 and 23

Even if Hattrup is available as reference, Applicants submit that Hong does not remedy the deficiencies in Hattrup.

1. Overview of Claims 1 and 23

Claim 1 recites:

> reading a log, wherein the log identifies a sequence in which blocks of data corresponding to multiple threads are stored on a sequential storage device;
> identifying at least a portion of the blocks of data as corresponding to one of the threads from the log; and
> indexing to the location of the identified portion of the blocks of data in the sequence in which blocks of data corresponding to multiple threads are stored on the sequential storage device according to the log

Claim 1 includes the features of i) identifying at least a portion of the blocks of data as corresponding to one of the threads from the log and ii) indexing to the location of the identified portion of the blocks of data in the sequence in which blocks data corresponding to multiple threads are stored on the sequential storage device according to the log; Claim 23 includes similar recitations.

According to claim 1, a device can identify the blocks of data as corresponding to one of the threads from the log. That is, for a particular thread, the device can identify which blocks of data belong to that thread from the log. Additionally, the device can index the locations of the identified blocks of data (i.e., the blocks of data that were identified from the log) in the sequence of intermingled blocks according to the log. Thus, the device can use the log to i) identify the blocks of data corresponding to a particular thread and ii) index the locations of the blocks of data. This allows the device to recover the blocks of data for a particular thread from intermingled blocks of data on a sequential storage device.

2. The Index of Hong is not the Log of Claims 1 and 23

a. Hong Does not Have a Log that Identifies a Sequence in Which Blocks of Data Corresponding to Multiple Threads are Stored"

Claim 1 recites "the log identifies a sequence in which blocks of data corresponding to multiple threads are stored." As is understood in the art, data is stored on many devices as blocks of data according to block protocols such as SCSI, for example. The log of Claims 1 and 23 identifies the sequence in which the blocks of data are stored on the sequential storage device.

Hong, on the other hand, discloses a method for inserting data records from multiple clients into a database. Data records from multiple buffers are merged into a single stream of records before being written to the database. *See,* Hong, Col. 8, Lines 22-27. An index can be created for the merged records before they are written to the database. *See,* Hong, Col. 3, lines 15-18; Col. 6, lines 44-49).

Hong builds an index of the records or files stored in the database. *See,* Hong, col. 6, line 50—col. 7, line 16. Other than describing that the index can be built, there is nothing in Hong that discusses the details of the index. While the index of Hong may contain some information about a record, there is nothing in Hong that teaches or suggests that index contains any information about the sequence in which records are stored. Moreover, even if Hong did discuss an index that records the sequence in which records are stored, which it does not, the analysis would not end there. Hong, is addressing efficient storage of the overall database records, and does not address storage and retrieval of the individual data blocks. Specifically, there is nothing in Hong that teaches or suggests that the index identifies a sequence in which blocks of data corresponding to multiple threads are stored.

b. Hong Does not Disclose "Identifying at Least a Portion of the Blocks of Data As Corresponding to One of the Threads from the Log"

Claims 1 and 23 recite "identifying at least a portion of the blocks of data as corresponding to one of the threads from the log." According to these claims, the blocks of data belonging to a particular thread can be identified from the log as corresponding to that thread. For example, if there are blocks of data from threads A, B, and C intermingled on a sequential storage media (e.g., a tape), the log can identify which blocks belong to thread A, which blocks belong to thread B and which blocks belong to thread C. The log can be created as the blocks of data are written to the sequential storage device. In one embodiment, "each entry in the log 20 includes and identifier of the thread to which the corresponding blocks belong."

Hong is concerned with storing data from multiple sources, not recovering data threads that are already stored. *See*, Hong, col. 2, lines 53-56. Moreover, Hong only discusses the index in general terms, but does not appear to discuss the contents of the index file. There is nothing in Hong that teaches or suggests that the index tracks which stream each record came from. Moreover, even if the index file of Hong did track such a thing, there is nothing that teaches or suggests that the log would allow for the identification of which data blocks, rather than records, belong to which thread. Thus, Hong does not teach or suggest "identifying at least a portion of the blocks of data as corresponding to one of the threads"

3. Hong Does Not Teach or Suggest "Indexing to the Location of the Identified Portion of the Blocks of Data in the Sequence In Which Blocks of Data Corresponding to Multiple Threads are Stored on the Sequential Storage Device According to the Log."

Claims 1 and 23 recite "indexing to the location of the identified portion of the blocks of data in the sequence in which blocks of data corresponding to multiple threads are stored on the sequential storage device according to the log." According to Claims 1 and 23, the present invention as recited can identify blocks corresponding to a thread from the log, as discussed above, and, for the identified blocks, index to the location of the identified blocks. This is described, for example, at ¶46 of the Specification, which states:

> When it is desired to retrieve a portion of the data recorded to the sequential storage device from multiple threads, it is first

> necessary to identify which thread and which block (or blocks) of that thread contains the desired data. That block can then be identified in the write log. (As noted above, all of the blocks of a given thread are recorded in order.) Once the block is identified in the write log, it is only necessary to index into the recorded sequence of blocks to locate the desired block. (For example, the number of blocks that precede the desired block in the recorded series of blocks can be determined and this number of blocks can be skipped from the beginning of the series.) The desired block can then be read.

Thus, when a block from a particular thread has been identified, the present invention, according to Claim 1, can index to the location of the identified block in the sequence of blocks according to the log. The step of indexing in Claim 1 is clearly not the step of initially creating the log file. Instead, the indexing step indexes to the location of blocks of data identified from the existing log (i.e., the identified portion of the blocks) as belonging to a particular thread.

It is unclear what the Examiner considers to be the indexing step as the Examiner simply states "Hong teaches sorting the data from a plurality of clients into a single data stream for storage . . . and creating an index of information equivalent to Applicant's log on the storage device." *See*, Feb. 2 Office Action, page 3. However, to the extent the Examiner equates the index of Hong to the log of Claim 1; the step of indexing in Hong is not the same as the indexing of the present invention. Indexing in Hong appears to be the process of building the index not a process of indexing locations of records identified from the already existing index. Thus, Hong does not teach or suggest identifying records belonging to a particular thread, and then, for the identified records, indexing to the locations of the records in a sequence of records according to the log file.

D. Claim 11

1. Overview of Claim 11

Claim 11 recites:

> A method for managing storage of blocks of data on a sequential storage device, wherein blocks of data corresponding to multiple threads are stored on the sequential storage device in an intermingled fashion, comprising:
> storing a sequence of blocks of data on a sequential storage device, wherein the blocks of data correspond to the multiple write threads and wherein blocks corresponding to different write

threads are intermingled on the sequential storage device;
recording the order in which the blocks of data are stored in a log; and
storing the log.

According to Claim 11, the method includes storing a sequence of blocks of data and recording the order in which the blocks of data are stored in a log. As is understood in the art, block protocols, such as SCSI, store blocks of data on storage devices on the block level. The method according to Claim 11 can record the order in which these data blocks are stored.

2. Hong Does not "Record the Order in Which the Blocks of Data are Stored in A Log"

As discussed above, Hong builds an index for a database record or records, not for the data blocks. Moreover, Hong fails to teach recording the order in which records, let alone data blocks, are stored.

D. Claim 16

1. Overview of Claim 16

Claim 16 recites:

A system for managing blocks of data on a sequential storage device, wherein blocks of data corresponding to multiple threads are stored on the sequential storage device in an intermingled fashion, comprising:
a sequential storage device configured to store intermingled blocks of data corresponding to the multiple threads;
a copy manager coupled to the sequential storage device and configured to manage the retrieval of copying of desired blocks of data from the sequential storage device; and
a memory coupled to the copy manager and configured to store a sequence in which blocks of data corresponding to the multiple threads are stored on the sequential storage device;
wherein the copy manager is configured to identify the position of the desired blocks of data in the sequence stored in the memory, to advance to a corresponding storage location on the sequential storage device without reading each of the preceding stored blocks of data or associated metadata, and to retrieve the desired blocks of data from the sequential storage device.

Claim 16 further recites a "copy manager" configured to "advance to a corresponding storage location on the sequential storage device without reading each of the preceding stored blocks of data or associated metadata." Thus, the copy manager can skip reading data and metadata for some of the proceeding blocks of data.

2. Hattrup Does Not Teach "To Advance to a Corresponding Storage Location on the Sequential Storage Device Without Reading Each of the Preceding Stored Blocks of Data or Associated Metadata"

In rejecting Claim 16, the Examiner stated that the copy manger of Hattrup "will count memory location equal to the offset to arrive at the identified data (Hattrup ¶87). This involved indexing to a location corresponding to identified data and does not involve reading all the preceding data."

As an initial matter, device 102 does not appear to be a copy manager (i.e., a device that handles extended copy commands for a host), but is part of a server that formats the data at server 102 or other data source so that the copy manager can move the data according to the extended copy command. Moreover, Claim 16 recites to advance to the corresponding storage location on the sequential storage device without reading . . ." not just indexing. Furthermore, ¶87 of Hattrup appears to disclose the use of markers of a portion of the metadata. The use of metadata, however, does not, in and of itself, suggest how previously saved data is read from a sequential target device. The '042 Application describes two methods for recovering data from a sequential target device. In the first, each of the data blocks is read and in the second the metadata for each of the data blocks is read. *See*, '042 Application, ¶49. The present invention allows this reading of the metadata (or data in the data block) to be skipped for blocks that are not being recovered. The use of metadata as described in Hattrup, consequently, does not require, nor does it suggest, to "advance to a corresponding storage location on the sequential storage device without reading each of the preceding stored blocks of data or associated metadata" as it is entirely possible that either the data or metadata for each preceeding block is read. Hong does not make up for this deficiency in Hattrup. Consequently, the rejection must fail.

E. Conclusion

For the foregoing reasons Applicants respectfully submit that Hattrup and Hong, alone or in combination, do not teach or suggest each of the limitations of the independent claims. Therefore, Applicants respectfully request allowance of Claims 1-23.

Applicant has now made an earnest attempt to place this case in condition for allowance. Other than as explicitly set forth above, this reply does not include an acquiescence to statements, assertions, assumptions, conclusions, or any combination thereof in the Office Action. For the foregoing reasons and for other reasons clearly apparent, Applicant respectfully requests full allowance of Claims 1-23. The Examiner is invited to telephone the undersigned at the number listed below for prompt action in the event any issues remain.

The Director of the U.S. Patent and Trademark Office is hereby authorized to charge any fees or credit any overpayments to Deposit Account No. 50-3183 of Sprinkle IP Law Group.

Respectfully submitted,

Sprinkle IP Law Group
Attorneys for Applicant

John L. Adair
Reg. No. 48,828

Date: June 19, 2006
1301 W. 25th Street, Suite 408
Austin, TX 78705
Tel. (512) 637-9223
Fax. (512) 371-9088